

Implementasi Metode Kontrol v, ω Berbasis Proporsional Integral Untuk Kontrol Gerak Mobile Robot Berpenggerak Differensial : Studi Simulasi

Ahmad Nashrul Aziz¹, Endra Pitowarno²
Jurusan Teknik Elektronika¹, Mekatronika²
Politeknik Elektronika Negeri Surabaya
Kampus ITS Keputih Sukolilo Surabaya 60111
Telp. (+62)31-5947280 Fax (+62) 31-5946114
Email: nashrul@nashrul.co.cc¹, epit@eepis-its.edu²

Abstrak: Pengembangan teknik otomasi pergerakan robot untuk dapat beroperasi di dunia nyata sudah menjadi bahan penelitian bagi pengembangan mobile robot di dunia saat ini. Untuk dapat mencapai suatu posisi yang diinginkan dari posisi awal, mobile robot membutuhkan suatu *sistem navigasi* yang dapat mengarahkan mobile robot tersebut ke posisi yang diinginkan. Arah perkembangan sistem navigasi mobile robot adalah menuju sebuah sistem navigasi yang tepat, dapat diandalkan, tidak mahal, dan tidak bergantung pada manusia demi efisiensi dan efektifitas kehidupan manusia.

Pada penelitian ini membahas tentang *kinematik* mobile robot berpenggerak differensial, *vo tracking control*. *vo tracking control* dilakukan simulasi terlebih dahulu dengan menggunakan Simulink MATLAB.

Kata kunci: Mobile robot, Kinematik, *vo tracking control*

1. Pendahuluan

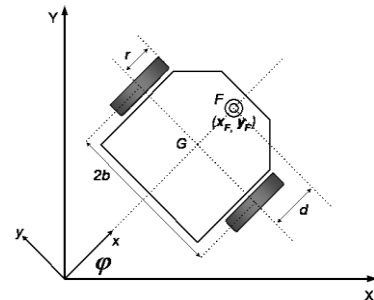
Salah satu jenis mobile robot yaitu mobile robot berpenggerak differensial atau *Differentially Driven Mobile Robot* (DDMR), adalah suatu mobile robot yang menggunakan dua buah roda penggerak yang independent, sehingga gerakan translasi maupun rotasi robot dihasilkan dari kombinasi pergerakan dua buah aktuator, supaya bisa stabil maka ditambah sebuah roda bebas (*omnidirectional*) yang biasa disebut roda *castor*.

Banyak metode kontrol robotik yang telah diperkenalkan seperti *Proportional-Integral-Derivative (PID) control*, *Resolve Acceleration Control (RAC)*, *adaptive control*, *active force control (AFC)*, *intelligent control*, *Iterative Learning Control (ILC)*, dan lain-lain. Hampir seluruh metode kontrol mengandung element klasik (yaitu PID kontrol) dimana terbukti bisa memberikan hasil yang lebih baik pada seluruh keandalan sistem. Salah satu kontrol robotik yang mengandung element PI kontrol adalah *vo control*.

Pada penelitian ini menggunakan *vo tracking control* untuk mengontrol pergerakan robot yang bersifat nonholonomik. Simulasi dari pergerakan dan kontrol robot menggunakan Simulink MATLAB.

1.1 Kinematik DDMR

Mobile robot yang dibahas pada penelitian ini diasumsikan bergerak pada bidang horizontal, serta berpenggerak dua buah roda kiri-kanan yang dikemudikan secara terpisah (*differentially driven* mobile robot, disingkat DDMR), seperti yang ditunjukkan dalam Gambar 1.



Gambar 1. DDMR pada medan 2D cartesian

Robot diasumsikan berada dalam kawasan 2D pada koordinat Cartesian XY. Parameter-parameter dalam gambar adalah:

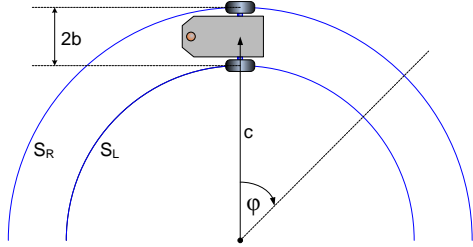
φ	sudut arah hadap robot
$2b$	lebar robot yang diukur dari garis tengah roda ke roda
r	jari-jari roda (roda kiri dan kanan adalah sama dan sebangun)
d	jarak antara titik tengah antara 2 roda, G dengan titik acuan F
(X_F, Y_F)	koordinat acuan di tubuh robot terhadap sumbu XY

Pertama-tama dicari nilai jarak tempuh roda kanan dan roda kiri dalam meter.

$$S_R = \text{Keliling roda} \cdot \frac{\text{Pulsa rotari kanan terukur}}{\text{Jumlah pulsa satu putaran roda}} \quad (1)$$

$$S_R = 2 \pi r \cdot \frac{\text{Pulsa rotari kanan terukur}}{\text{Jumlah pulsa satu putaran roda}} \quad (2)$$

$$S_L = 2 \pi r \cdot \frac{\text{Pulsa rotari kiri terukur}}{\text{Jumlah pulsa satu putaran roda}} \quad (3)$$



Gambar 2. Lintasan yang dilalui DDMR

Dengan cara membagi jumlah pulsa rotari yang terukur dibagi jumlah pulsa satu putaran roda akan didapatkan nilai putaran roda, nilai ini akan dikalikan dengan keliling roda yang bersangkutan. Untuk mencari jarak tempuh robot bisa dicari dengan.

$$S = \frac{S_L + S_R}{2} \quad (4)$$

Persamaan (4) dipakai untuk pergerakan maju, mundur, atau berputar ditempat (*Pivot*), selanjutnya harus diketahui sudut rotasi robot selama menempuh jarak S . Diasumsikan robot bergerak melalui bagian lintasan yang berbentuk lingkaran. S_L dan S_R didefinisikan sebagai dua buah busur lingkaran yang mana nilai sudut yang ditempuh adalah (ϕ dalam radian). Busur S_L dan S_R dapat dicari dengan cara mengalikan sudut tempuh ϕ , dengan jari-jari lintasan tiap roda. Jika jari-jari pusat perputaran robot adalah c , maka selama robot berputar ke kiri, jari-jari lintasan roda kanan adalah

$$\frac{c + 2b}{2} \quad (5)$$

serta jari-jari lintasan roda kiri adalah

$$\frac{c - 2b}{2} \quad (6)$$

kedua busur memiliki titik pusat yang sama.

$$S_R = \phi \frac{c + 2b}{2} \quad (7)$$

$$S_L = \phi \frac{c - 2b}{2} \quad (8)$$

Apabila kedua persamaan diatas dikurangkan, maka c akan hilang.

$$S_R - S_L = \phi \cdot 2b \quad (9)$$

Dan akhirnya didapatkan nilai ϕ , yaitu:

$$\phi = \frac{S_R - S_L}{2b} \quad (10)$$

Dengan menggunakan kecepatan linier roda V_L, V_R yang menyebabkan robot mencapai jarak S_L, S_R dan

kecepatan angular $\dot{\theta}_L, \dot{\theta}_R$ (dalam radian per detik), serta jari-jari roda r , akan didapatkan [1].

$$\dot{\theta}_R = \frac{1}{2\pi r} V_R \quad (11)$$

$$\dot{\theta}_L = \frac{1}{2\pi r} V_L \quad (12)$$

Dengan mengabaikan analisis dari castor bebas, konfigurasi dari platform dapat dideskripsikan kedalam lima variabel umum (q). Tiga variabel menggambarkan posisi robot, dua variabel adalah posisi sudut kedua roda. Sehingga koordinat umum lagrange (q) dan input kontrol (u) adalah,

$$q = (x, y, \theta, \theta_R, \theta_L) \quad (13)$$

$$u = (\dot{\theta}_R, \dot{\theta}_L)^T \text{ atau } u = \begin{pmatrix} \dot{\theta}_R \\ \dot{\theta}_L \end{pmatrix} \quad (14)$$

dimana (X_F, Y_F) adalah koordinat terhadap pusat gravitasi, $\dot{\theta}_R$ dan $\dot{\theta}_L$ adalah kecepatan sudut roda kanan dan roda kiri.

Sifat nonholonomik mempunyai keterbatasan dalam hal gerakan. Fungsi geometri tertentu yang berhubungan arah hadap harus dipenuhi untuk mendapatkan gerak yang sesuai. Diasumsikan roda tidak mengalami slip, maka ada tiga konstrain yang bisa dinyatakan. Pertama, kecepatan pada titik X_F, Y_F terhadap medan sumbu X dan sumbu Y adalah,

$$\dot{y} \cos \theta - \dot{x} \sin \theta = 0 \quad (15)$$

selanjutnya, jika aktuasi roda dianggap tidak slip, kemudian mengikuti persamaan sebelumnya, maka akan didapatkan

$$\dot{y} \cos \theta + \dot{x} \sin \theta + b \dot{\theta} = r \dot{\theta}_R \quad (16)$$

$$\dot{y} \cos \theta + \dot{x} \sin \theta - b \dot{\theta} = r \dot{\theta}_L \quad (17)$$

dimana

$$\dot{x} = v \cos \theta \quad (18)$$

$$\dot{y} = v \sin \theta \quad (19)$$

$$\dot{\theta} = \omega \quad (20)$$

besaran (v, ω) mengacu pada kecepatan linier dan kecepatan sudut. Persamaan (15) sampai (17) dapat ditulis ulang dalam bentuk yang digambarkan pada [2], [3].

$$A(q) \dot{q} = 0 \quad (21)$$

dimana

$$A(q) = \begin{bmatrix} \sin \theta & -\cos \theta & 0 & 0 & 0 \\ \cos \theta & \sin \theta & b & -r & 0 \\ \cos \theta & \sin \theta & -b & 0 & -r \end{bmatrix} \quad (22)$$

sebuah kendaraan beroda dengan gerakan nonholonomik, kinematiknya dapat ditulis sebagai berikut,

$$\dot{q} = S(q)u(t) \quad (23)$$

dimana $S(q)$ ditunjukkan seperti,

$$S(q) = \begin{bmatrix} \frac{r}{2} \cos \theta & \frac{r}{2} \cos \theta \\ \frac{r}{2} \sin \theta & \frac{r}{2} \sin \theta \\ \frac{r}{2b} & -\frac{r}{2b} \\ 0 & 1 \end{bmatrix} \quad (24)$$

menggunakan persamaan (24), itu adalah kinematik maju dari pergerakan robot. Kinematik maju digunakan untuk memperkirakan posisi dan kecepatan dalam ruang cartesian. Dengan mengasumsikan kecepatan sudut, (u) sebagai input, model kinematik dari gerakan sebagai berikut,

$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ \theta_R \\ \theta_L \end{bmatrix} = \begin{bmatrix} \frac{r}{2} \cos \theta & \frac{r}{2} \cos \theta \\ \frac{r}{2} \sin \theta & \frac{r}{2} \sin \theta \\ \frac{r}{2b} & -\frac{r}{2b} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (25)$$

dimana u_1 dan u_2 merepresentasikan kecepatan sudut kiri dan kanan roda. kemudian hubungan antara v, ω dan u_1, u_2 adalah,

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & \frac{b}{r} \\ \frac{1}{r} & -\frac{b}{r} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (26)$$

mensubstitusikan persamaan (26) kedalam persamaan (25), difokuskan hanya pada tiga keadaan x, y dan θ (termasuk θ_R, θ_L), persamaan umum kecepatan sebuah DDMR dapat dinyatakan sebagai,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (27)$$

sistem DDMR yang digambarkan oleh persamaan (25) sampai (27) disebut *steering system*.

1.2 Dinamik DDMR

Dengan mengasumsikan massa robot adalah m_c , massa roda dan motor adalah m_w . I_c, I_w, I_m adalah moment inersia bodi robot terhadap sumbu vertical pada titik F, roda dengan motor terhadap sumbu roda dan roda dengan motor terhadap diameter roda. Persamaan model dinamik DDMR didefinisikan pada persamaan berikut, [2], [3].

$$M(q)\dot{v} + V(q, \dot{q}) = B(q)\tau \quad (28)$$

dimana $\tau = [\tau_r, \tau_l]$ adalah torsi yang diberikan pada roda kanan dan roda kiri, $M(q), V(q, \dot{q}), B(q)$ dirumuskan sebagai berikut:

$$M(q) = \begin{bmatrix} \frac{r^2}{4b^2}(m_b^2 + I) + I_w & \frac{r^2}{4b^2}(m_b^2 - I) \\ \frac{r^2}{4b^2}(m_b^2 - I) & \frac{r^2}{4b^2}(m_b^2 + I) + I_w \end{bmatrix} \quad (29)$$

$$V(q, \dot{q}) = \begin{bmatrix} 0 & \frac{r^2}{2b} m_c d \dot{\phi} \\ -\frac{r^2}{2b} m_c d \dot{\phi} & 0 \end{bmatrix} \quad (30)$$

$$B(q) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (31)$$

I, m adalah momen inersia total, massa total robot, dapat dicari dengan menggunakan persamaan $I = m_c d^2 + 2m_w b^2 + I_c + 2I_m$, dan $m = m_c + 2m_w$.

1.3 Penelusuran Trajektori (Trajectory Tracking)

Tugas yang sulit dalam pengaturan gerak robot adalah pada bagian pengaturan kecepatan input roda, untuk menelusuri trajektori referensi yang telah di generate sebelumnya (*trajectory tracking*). Pertama-tama yang harus dilakukan yaitu mengatur error posisi, dengan mengasumsikan posisi referensi $q_{ref} = (x_{ref}, y_{ref}, \phi_{ref})^T$, dan posisi aktual robot adalah $q_{act} = (x_{act}, y_{act}, \phi_{act})^T$. Error posisi $q_e = (e_1, e_2, e_3)^T$ adalah transformasi dari posisi referensi q_{ref} pada sebuah sistem koordinat lokal dengan posisi asal (x_{act}, y_{act}) dan sumbu x dengan arah hadap robot adalah ϕ_{act} . Sehingga dapat diwujudkan dalam persamaan matematis, sebagai berikut,

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos \phi_{act} & \sin \phi_{act} & 0 \\ -\sin \phi_{act} & \cos \phi_{act} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{ref} - x_{act} \\ y_{ref} - y_{act} \\ \phi_{ref} - \phi_{act} \end{bmatrix} \quad (32)$$

Model referensi kecepatan linier dan angular (v, ω) yang bisa menyebabkan q_e konvergen ke nol, diberikan oleh [4], [5], adalah sebagai berikut,

$$v_{des} = v_{ref} \cos e_3 + K_1 e_1 \quad (33)$$

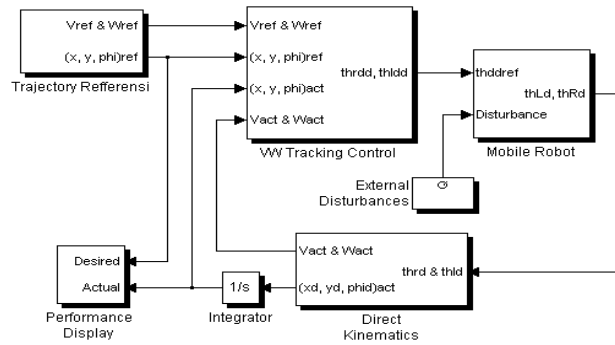
$$\omega_{des} = \omega_{ref} + v_{ref} K_2 e_2 + K_3 \sin e_3 \quad (34)$$

Dengan K_1, K_2, K_3 adalah konstanta positif.

2. Simulasi Pergerakan Robot

Model simulasi dibuat dengan menggunakan Simulink Matlab. Simulasi yang dibuat dalam Matlab nantinya berkaitan dengan bagian kinematik, dinamik robot, serta yang berkaitan dengan metode kontrol yang dipakai. Metode yang diajukan adalah skema kontrol kecepatan menggunakan *vdw control*. Dinamik robot

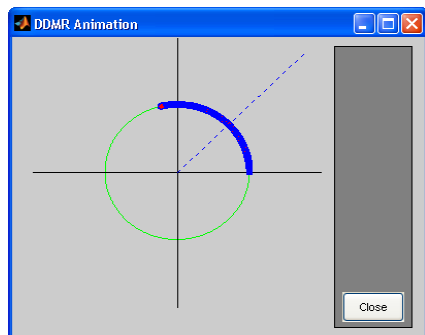
dibuat dalam blok-blok simulasi dengan mengasumsikan gangguan yang terjadi. Pada proses simulasi yaitu memakai *solver ode45(Dorman-Prince)*, tipe *Variable-step*. Blok diagram lengkap dari simulasi robot diperlihatkan pada gambar dibawah ini,



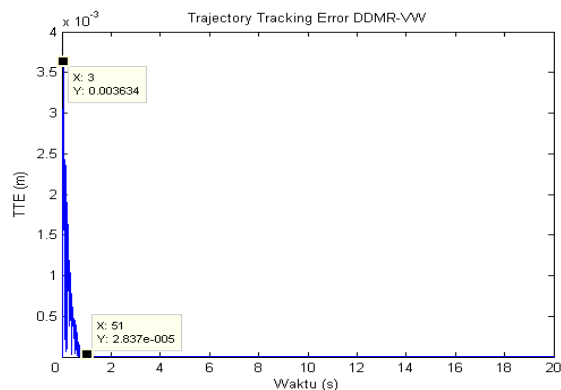
Gambar 3: Blok simulasi DDMR dengan memakai *vw tracking control*

3. Data Simulasi dan Analisa Hasil

Pada proses simulasi akan dilakukan tuning terhadap konstanta kontrol serta menyesuaikan parameter yang ada pada fisik robot, kecuali pada kotak dialog gangguan luar. Pada bagian ini akan ditampilkan hasil simulasi dalam bentuk trajectory tracking error (TTE) dengan *vw tracking control*, yang dimaksud dengan TTE, adalah selisih antara trajectory yang dikehendaki dengan trajectory yang sebenarnya (actual), semakin besar nilai TTE, artinya posisi atau kecepatan aktual semakin jauh dari posisi atau kecepatan yang dikehendaki.

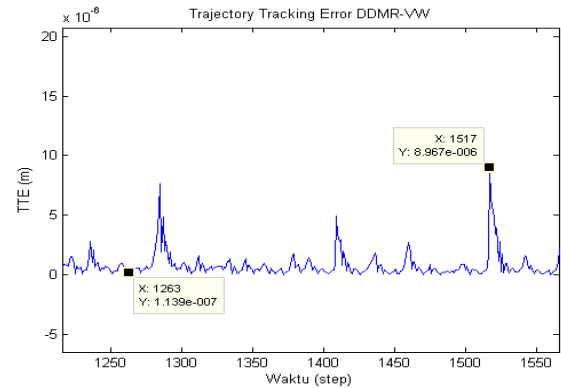


Gambar 4: Tampilan animasi DDMR



Gambar 5: Hasil monitoring track error tanpa gangguan luar

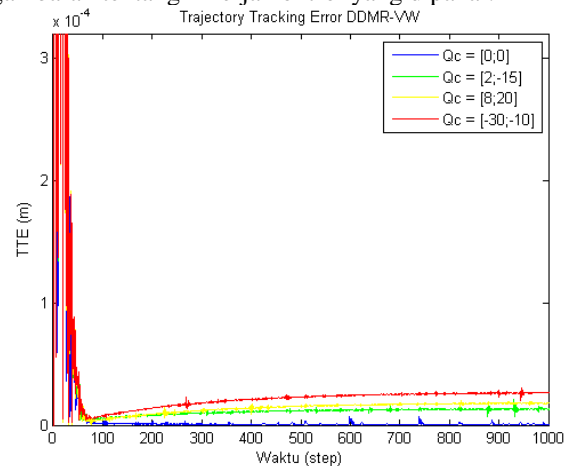
Setelah dilakukan *running* maka akan terlihat fungsi animasi berjalan sesuai dengan trajectory yang telah ditentukan, dengan waktu simulasi selama 20 (s), inisial posisi ($x=10$, $y=0$, $\phi=\pi/2$), tipe trajectory lingkaran, jari-jari lingkaran sebesar 10 (m), kecepatan sebesar 0.6 (m/s), dan tanpa memberikan gangguan luar, tampilan animasi dari simulasi diperlihatkan pada gambar 4.



Gambar 6: Osilasi track error pada kondisi tanpa gangguan luar

Hasil monitoring track error pada kondisi tanpa gangguan luar diperlihatkan pada gambar 5, disitu bisa dilihat bahwa kinerja kontroler memberikan hasil yang cukup baik dengan nilai steady state error dibawah 2.837×10^{-5} meter, terjadi overshoot maksimal pada step ke 3 sebesar 3.634×10^{-3} meter, sedangkan track error mulai menunjukkan kestabilan mulai pada step ke 51. Osilasi track error bisa dilihat pada gambar 6. Pada keadaan steady state terjadi osilasi, dengan nilai puncak sebesar 8.967×10^{-6} meter pada step ke 1517, nilai terendah sebesar 1.139×10^{-7} terjadi pada step ke 1263.

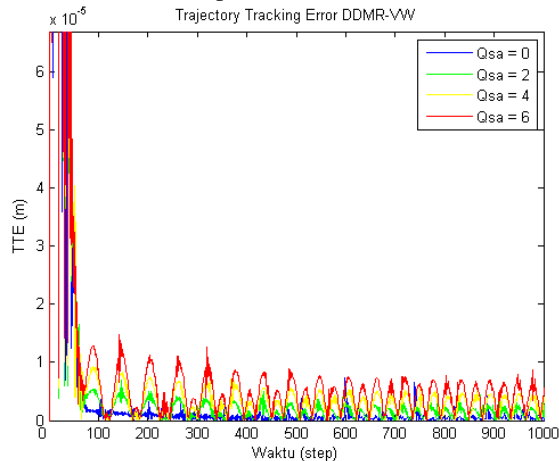
Berikutnya dilakukan simulasi dengan memberikan gangguan luar menggunakan tipe gangguan dan nilai yang berubah-ubah, ini dilakukan untuk mengetahui performa kontrol yang digunakan, sehingga pada proses eksperimen telah mendapatkan gambaran tentang kinerja kontrol yang dipakai.



Gambar 7: Hasil monitoring track error pada kondisi diberi gangguan konstan

Pada gambar 7 diperlihatkan hasil monitoring track error yaitu dengan memberikan gangguan konstan, dilakukan 4 kali simulasi dengan nilai konstanta (Q_c) sebagai berikut,

1. Simulasi pertama, $Q_c = [0;0]$
2. Simulasi kedua, $Q_c = [2;-15]$
3. Simulasi ketiga, $Q_c = [8;20]$
4. Simulasi keempat, $Q_c = [-30;-10]$



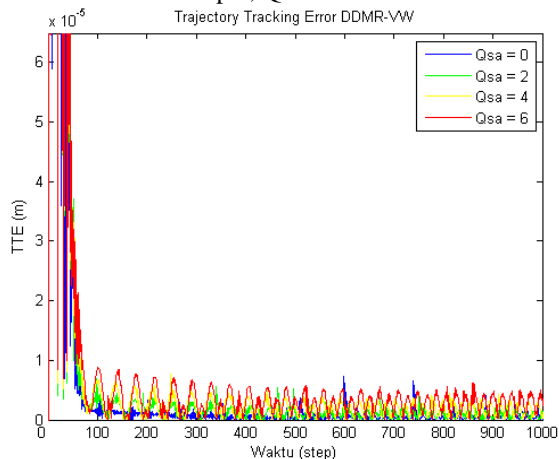
Gambar 8: Hasil monitoring track error pada kondisi diberi gangguan *sine wave* dengan frekuensi 10 Hz

Pada gambar 8 diperlihatkan hasil monitoring track error yaitu dengan memberikan gangguan *sine wave* (Q_{sf}) sebesar 10 Hz, dilakukan 4 kali simulasi dengan nilai konstanta (Q_{sa}) sebagai berikut,

1. Simulasi pertama, $Q_{sa} = 0$
2. Simulasi kedua, $Q_{sa} = 2$
3. Simulasi ketiga, $Q_{sa} = 4$
4. Simulasi keempat, $Q_{sa} = 6$

Pada gambar 9 diperlihatkan hasil monitoring track error yaitu dengan memberikan gangguan *sine wave* (Q_{sf}) sebesar 20 Hz, dilakukan 4 kali simulasi dengan nilai konstanta (Q_{sa}) sebagai berikut,

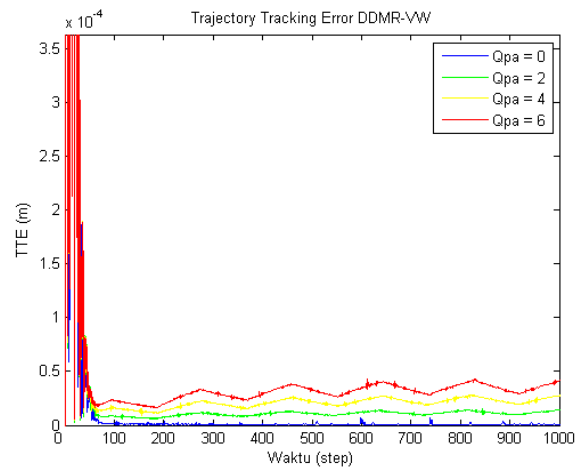
1. Simulasi pertama, $Q_{sa} = 0$
2. Simulasi kedua, $Q_{sa} = 2$
3. Simulasi ketiga, $Q_{sa} = 4$
4. Simulasi keempat, $Q_{sa} = 6$



Gambar 9: Hasil monitoring track error pada kondisi diberi gangguan *sine wave* dengan frekuensi 20 Hz

Pada gambar 10 diperlihatkan hasil monitoring track error yaitu dengan memberikan gangguan *pulse* (Q_{pf}) sebesar 4 Hz, dilakukan 4 kali simulasi dengan nilai konstanta (Q_{pa}) sebagai berikut,

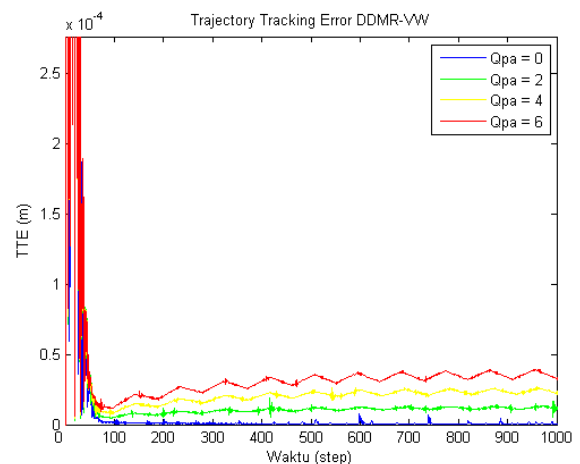
1. Simulasi pertama, $Q_{pa} = 0$
2. Simulasi kedua, $Q_{pa} = 2$
3. Simulasi ketiga, $Q_{pa} = 4$
4. Simulasi keempat, $Q_{pa} = 6$



Gambar 10: Hasil monitoring track error pada kondisi diberi gangguan *pulse* dengan frekuensi 4 Hz

Pada gambar 11 diperlihatkan hasil monitoring track error yaitu dengan memberikan gangguan *pulse* (Q_{pf}) sebesar 8 Hz, dilakukan 4 kali simulasi dengan nilai konstanta (Q_{pa}) sebagai berikut,

1. Simulasi pertama, $Q_{pa} = 0$
2. Simulasi kedua, $Q_{pa} = 2$
3. Simulasi ketiga, $Q_{pa} = 4$
4. Simulasi keempat, $Q_{pa} = 6$



Gambar 11: Hasil monitoring track error pada kondisi diberi gangguan *pulse* dengan frekuensi 8 Hz

4. Kesimpulan

Dari seluruh proses yang telah dibahas pada penelitian ini bisa diperoleh kesimpulan sebagai berikut:

1. Pemakaian komponen kontrol PI (Proporsional Integral) mempunyai pengaruh baik terhadap *vo tracking control*, pada hasil simulasi, yaitu mampu meredam *overshoot* dan juga mengurangi osilasi serta *steady state error* menjadi lebih kecil. Hal ini menunjukkan bahwa pemakaian komponen I mampu menjadikan performa *vo tracking control* menjadi lebih *robust* terhadap gangguan, yaitu kemampuan mengkompensasi gerak terhadap gangguan luar.

5. Referensi

- [1] Braunl T. (2003). *Embedded Robotics - Mobile Robot Design and Applications with Embedded Systems*. Buku Teks. Berlin: Springer.
- [2] Fukao, T., Nakagawa, H., and Adachi, N. (2000). *Adaptive Tracking Control of a Nonholonomic Mobile Robot*. IEEE Transactions Robotot and Automation. 16(5). 609-615.
- [3] Fierro, R. and Lewis, F. L. (1995). *Control of Nonholonomic Mobile Robot: Backstepping Kinematics into Dynamics*. Proc. 34th Conf. On Decision and Control. 3805-3810.
- [4] Kolmanovsky I., and McClamroch, N. H. (1995). *Development in Nonholonomic Control Problems*. IEEE Control Systems, pp. 20–36.
- [5] Fierro, R., and Lewis, F. L. (1998). *Control of a Nonholonomic Mobile Robot Using Neural Networks*. IEEE Trans. on neural networks, vol. 9, pp. 589–600.